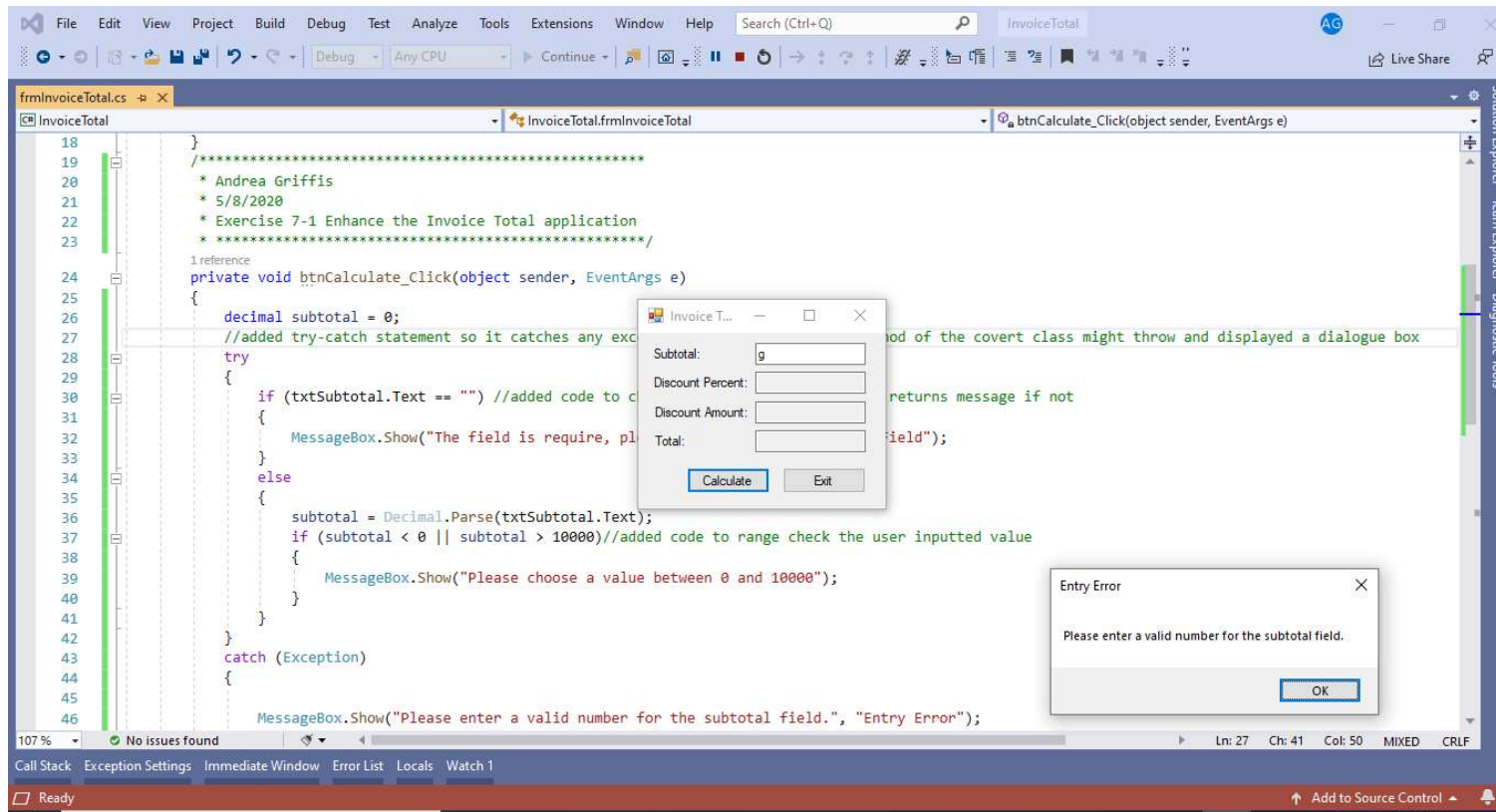


Assignment: Week 5 Practice Assignments (Individual)

Ex.7-1 Enhance the Invoice Total application

(#2)



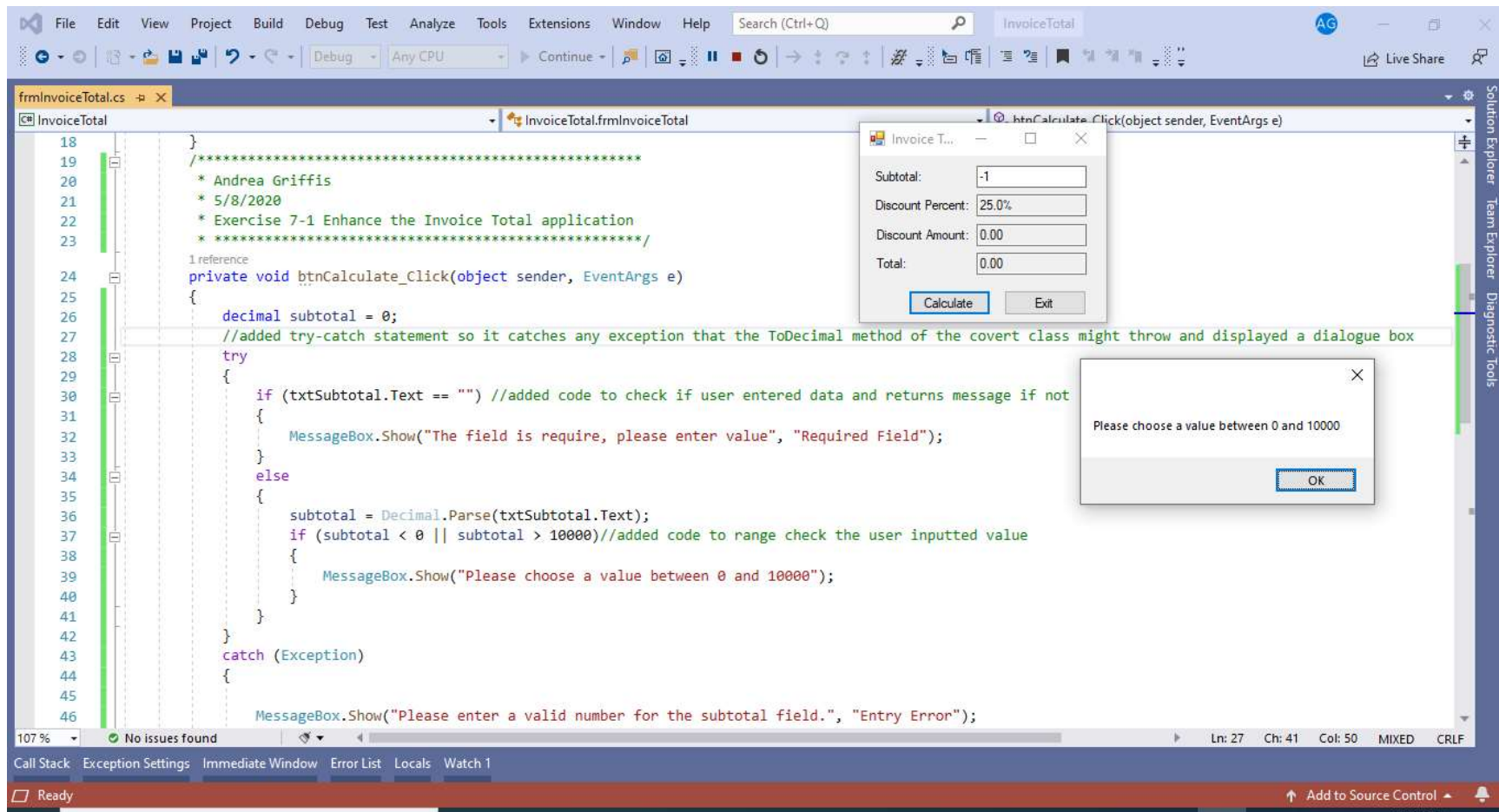
(#3)

The screenshot displays the Visual Studio IDE with the following components:

- Code Editor:** Shows the `frmInvoiceTotal.cs` file. The code includes a comment block with author information and a `btnCalculate_Click` method. The method calculates the subtotal, checks for required input, and validates the range (0 to 10000).

```
18 }
19
20 /*****
21  * Andrea Griffis
22  * 5/8/2020
23  * Exercise 7-1 Enhance the Invoice Total application
24  * *****/
25
26 1 reference
27 private void btnCalculate_Click(object sender, EventArgs e)
28 {
29     decimal subtotal = 0;
30     //added try-catch statement so it catches any exception that the.ToDecimal method of the covert class might throw and displayed a dialogue box
31     try
32     {
33         if (txtSubtotal.Text == "") //added code to check if user entered data and returns message if not
34         {
35             MessageBox.Show("The field is require, please enter value", "Required Field");
36         }
37         else
38         {
39             subtotal = Decimal.Parse(txtSubtotal.Text);
40             if (subtotal < 0 || subtotal > 10000) //added code to range check the user inputted value
41             {
42                 MessageBox.Show("Please choose a value between 0 and 10000");
43             }
44         }
45     }
46     catch (Exception)
47     {
48         MessageBox.Show("Please enter a valid number for the subtotal field.", "Entry Error");
49     }
50 }
```
- Form Designer:** Shows a preview of the `InvoiceTotal` form with fields for Subtotal, Discount Percent (25.0%), Discount Amount (0.00), and Total (0.00). Buttons for Calculate and Exit are visible.
- Message Boxes:** Two dialog boxes are shown. The first, titled "Required Field", displays the message "The field is require, please enter value". The second, titled "Entry Error", displays the message "Please enter a valid number for the subtotal field.".
- Interface:** The top menu bar includes File, Edit, View, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, and Help. The bottom status bar shows "Ready" and "Add to Source Control".

(#4)



Ex. 7-2 Enhance the Future Value application

(#2)

The screenshot shows the Visual Studio IDE with a C# code file open. The code is for a 'Future Value' application. It includes a 'CalculateFutureValue' method that takes 'monthlyInvestment' and 'monthlyInterestRate' as parameters. The code calculates the future value based on these inputs and the number of years. A 'Format Entry Error' dialog box is displayed, indicating a format exception has occurred. The dialog box has an 'OK' button. The code file also shows a 'btnCalculate_Click' event handler that calls the 'CalculateFutureValue' method. The IDE interface includes a menu bar, a toolbar, and a Solution Explorer on the right.

```
19  /*****  
20  * Andrea Griffis  
21  * 5/8/2020  
22  * Exercise 7-2 Enhance the Future Value application  
23  * *****/  
24  1 reference  
25  private void btnCalculate_Click(object sender, EventArgs e)  
26  {  
27      try {  
28          decimal monthlyInvestment = Convert.ToDecimal(txtMonthlyInvestment.Text);  
29          decimal yearlyInterestRate = Convert.ToDecimal(txtInterestRate.Text);  
30          int years = Convert.ToInt32(txtYears.Text);  
31  
32          int months = years * 12;  
33          decimal monthlyInterestRate = yearlyInterestRate / 12 / 100;  
34  
35          decimal futureValue = this.CalculateFutureValue(  
36              monthlyInvestment, monthlyInterestRate, months);  
37          txtFutureValue.Text = futureValue.ToString("c");  
38          txtMonthlyInvestment.Focus();  
39      }  
40      // added catch for format exception  
41      catch (FormatException)  
42      {  
43          MessageBox.Show("A format exception has occurred, Please check all entries.",  
44              "Format Entry Error");  
45      }  
46      //added catch for overflow exception  
47      catch (OverflowException)  
48      {  
49      }  
50  }
```

Future Value

Monthly Investment: xx
Yearly Interest Rate:
Number of Years:
Future Value:
Calculate Exit

Format Entry Error

A format exception has occurred, Please check all entries.
OK

Visual Studio interface showing the development of a Future Value application. The code in Form1.cs [Design] is as follows:

```
19  /*****  
20  * Andrea Griffis  
21  * 5/8/2020  
22  * Exercise 7-2 Enhance the Future Value application  
23  * *****/  
24  1 reference  
25  private void btnCalculate_Click(object sender, EventArgs e)  
26  {  
27      try {  
28          decimal monthlyInvestment = Convert.ToDecimal(txtMonthlyInvestment.Text);  
29          decimal yearlyInterestRate = Convert.ToDecimal(txtInterestRate.Text);  
30          int years = Convert.ToInt32(txtYears.Text);  
31  
32          int months = years * 12;  
33          decimal monthlyInterestRate = yearlyInterestRate / 12 / 100;  
34  
35          decimal futureValue = this.CalculateFutureValue(  
36              monthlyInvestment, monthlyInterestRate, months);  
37          txtFutureValue.Text = futureValue.ToString("c");  
38          txtMonthlyInvestment.Focus();  
39      }  
40      // added catch for format exception  
41      catch (FormatException)  
42      {  
43          MessageBox.Show("A format exception has occurred, Please check all entries.",  
44              "Format Entry Error");  
45      }  
46      //added catch for overflow exception  
47      catch (OverflowException)  
48      {
```

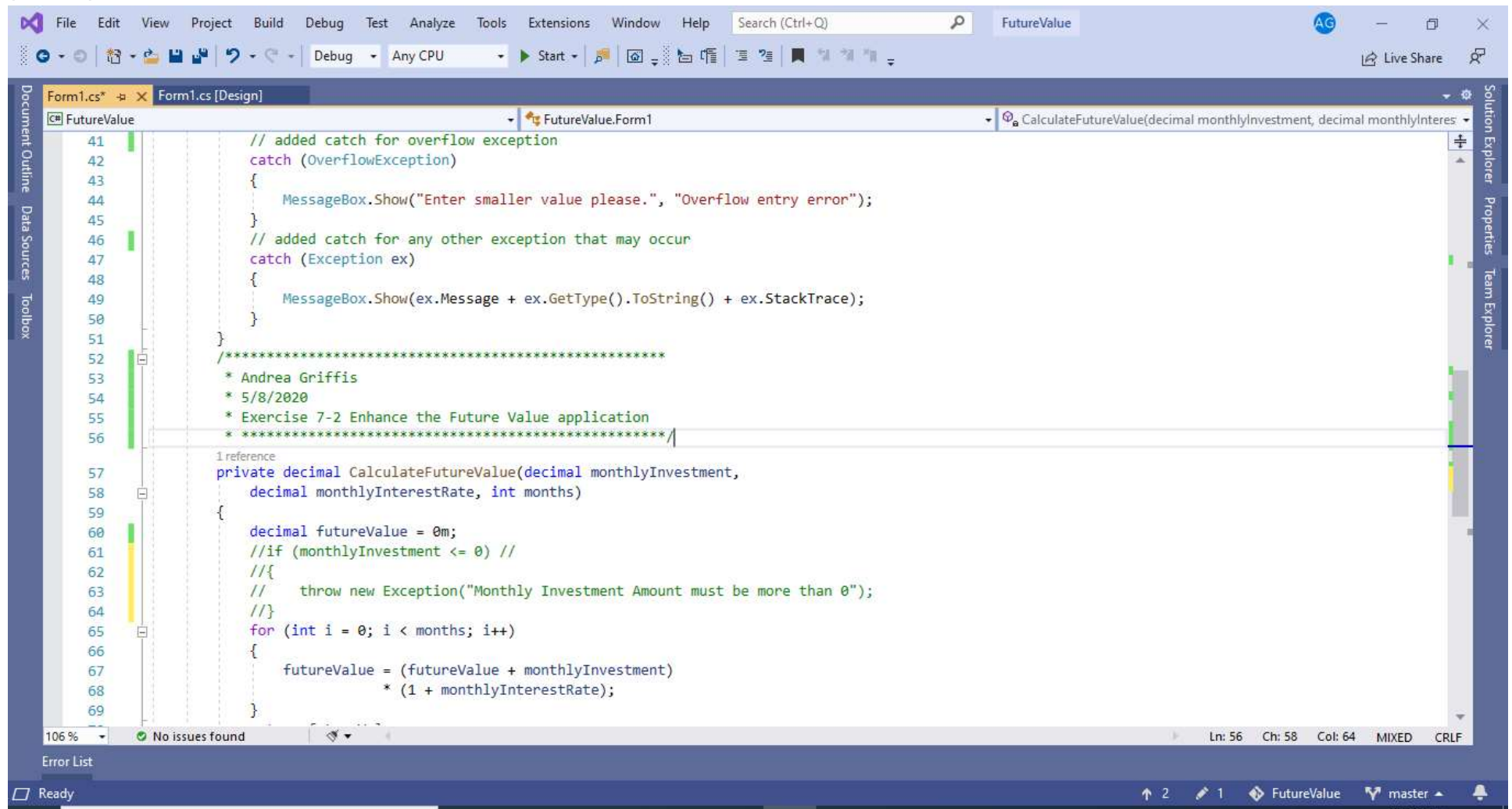
The application window "Future Value" displays the following input fields and buttons:

- Monthly Investment: 1000
- Yearly Interest Rate: 10
- Number of Years: 5000000000000000
- Future Value:
- Buttons: Calculate, Exit

An "Overflow entry error" dialog box is displayed, indicating that the input value for the number of years is too large. The message says: "Enter smaller value please." with an OK button.

The status bar at the bottom shows "107 %", "No issues found", and "Ln: 54 Ch: 53 MIXED CRLF".

(#3 & 4)



```
41 // added catch for overflow exception
42 catch (OverflowException)
43 {
44     MessageBox.Show("Enter smaller value please.", "Overflow entry error");
45 }
46 // added catch for any other exception that may occur
47 catch (Exception ex)
48 {
49     MessageBox.Show(ex.Message + ex.GetType().ToString() + ex.StackTrace);
50 }
51 }
52 /*****
53  * Andrea Griffis
54  * 5/8/2020
55  * Exercise 7-2 Enhance the Future Value application
56  * *****/
57 1 reference
58 private decimal CalculateFutureValue(decimal monthlyInvestment,
59     decimal monthlyInterestRate, int months)
60 {
61     decimal futureValue = 0m;
62     //if (monthlyInvestment <= 0) //
63     //{
64         throw new Exception("Monthly Investment Amount must be more than 0");
65     //}
66     for (int i = 0; i < months; i++)
67     {
68         futureValue = (futureValue + monthlyInvestment)
69             * (1 + monthlyInterestRate);
70     }
71 }
```

106 % No issues found Ln: 56 Ch: 58 Col: 64 MIXED CRLF

Ready

(#5)

Visual Studio interface showing the code editor for `Form1.cs` in the `FutureValue` project. The code defines two methods: `IsPresent` and `IsDecimal`.

```
86  /*****  
87  * Andrea Griffis  
88  * 5/8/2020  
89  * Exercise 7-2 Enhance the Future Value application  
90  * *****/  
91  
92  3 references  
93  public bool IsPresent(TextBox textBox, string name)  
94  {  
95      if (textBox.Text == "")  
96      {  
97          MessageBox.Show(name + " is a required field.", "Entry Error");  
98          textBox.Focus();  
99          return false;  
100     }  
101     return true;  
102  
103  3 references  
104  public bool IsDecimal(TextBox textBox, string name)  
105  {  
106      decimal number = 0m;  
107      if (Decimal.TryParse(textBox.Text, out number))  
108      {  
109          return true;  
110      }  
111      else  
112      {  
113          MessageBox.Show(name + " must be a decimal value.", "Entry Error");  
114          textBox.Focus();  
115          return false;  
116      }  
117  }
```

The interface includes the menu bar (File, Edit, View, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help), the toolbar, the Document Outline, Data Sources, and Toolbox on the left. The Solution Explorer, Team Explorer, and Properties windows are on the right. The status bar at the bottom shows "106%", "No issues found", and "Ln: 90 Ch: 58 Col: 64 MIXED CRLF".

File Edit View Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) FutureValue

Debug Any CPU Start

Form1.cs* Form1.cs [Design]* FutureValue FutureValue.Form1 IsInt32(TextBox textBox, string name)

```
117 * Andrea Griffis
118 * 5/8/2020
119 * Exercise 7-2 Enhance the Future Value application
120 * *****/
121 0 references
122 public bool IsInt32(TextBox textBox, string name)
123 {
124     int number = 0;
125     if (Int32.TryParse(textBox.Text, out number))
126     {
127         return true;
128     }
129     else
130     {
131         MessageBox.Show(name + " must be a integer.", "Entry Error");
132         textBox.Focus();
133         return false;
134     }
135 3 references
136 public bool IsWithinRange(TextBox textBox, string name, decimal min, decimal max)
137 {
138     decimal number = Convert.ToDecimal(textBox.Text);
139     if (number < min || number > max)
140     {
141         MessageBox.Show(name + " must be between " + min.ToString() + " and " + max.ToString() + ".", "Entry Error");
142         textBox.Focus();
143         return false;
144     }
145     return true;
146 }
```

106% No issues found Ln: 134 Ch: 10 MIXED CRLF

Error List

Ready 2 1 FutureValue master

(#6)

The screenshot shows the Visual Studio IDE with the 'FutureValue' project open. The 'Form1.cs [Design]*' file is selected, and the 'FutureValue' namespace is chosen. The code editor displays the following C# code:

```
68  /*****  
69  * Andrea Griffis  
70  * 5/8/2020  
71  * Exercise 7-2 Enhance the Future Value application  
72  * *****/  
73  
74  0 references  
75  public bool IsValid()  
76  {  
77      return  
78          // validate monthly investment textbox  
79          IsPresent(txtMonthlyInvestment, "Monthly Investment") &&  
80          IsDecimal(txtMonthlyInvestment, "Monthly Investment") &&  
81          IsWithinRange(txtMonthlyInvestment, "Monthly Investment", 1, 1000) &&  
82          // validate interest rate text box  
83          IsPresent(txtInterestRate, "Investment Rate") &&  
84          IsDecimal(txtInterestRate, "Investment Rate") &&  
85          IsWithinRange(txtInterestRate, "Investment Rate", 1, 20) &&  
86          // validate Number of Years textbox  
87          IsPresent(txtYears, "Number of years") &&  
88          IsDecimal(txtYears, "Number of years") &&  
89          IsWithinRange(txtYears, "Number of years", 1, 40);  
90  }  
91  
92  3 references  
93  public bool IsPresent(TextBox textBox, string name)  
94  {  
95      if (textBox.Text == "")  
96      {  
97          MessageBox.Show(name + " is a required field.", "Entry Error");  
98      }  
99  }
```

The status bar at the bottom indicates '106 %', 'No issues found', and the current cursor position is 'Ln: 72, Ch: 58, Col: 64, MIXED, CRLF'. The 'Error List' pane is empty.

(#7)

The screenshot shows the Visual Studio IDE with the following components:

- Menu Bar:** File, Edit, View, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q) with a magnifying glass icon.
- Toolbox:** Debug, Any CPU, Start, and various icons for running and debugging.
- Document Outline:** Shows the project structure with 'FutureValue' and 'Form1.cs [Design]*'.
- Code Editor:** Displays the code for 'btnCalculate_Click(object sender, EventArgs e)'. The code includes comments about the author (Andrea Griffis) and date (5/8/2020), and implements a try-catch block for data validation and calculation.
- Properties Window:** Shows the 'btnCalculate_Click' event handler.
- Output Window:** Shows '1 reference' for the event handler.
- Bottom Bar:** Includes a status bar with '106 %', 'No issues found', and a footer with 'Ln: 23 Ch: 58 Col: 64 MIXED CRLF'.

```
18 }
19 /*****
20  * Andrea Griffis
21  * 5/8/2020
22  * Exercise 7-2 Enhance the Future Value application
23  *****/
24 private void btnCalculate_Click(object sender, EventArgs e)
25 {
26     try
27     {
28         if (IsValidData()) // modified this event handler to validate data before it processes it
29         {
30             decimal monthlyInvestment = Convert.ToDecimal(txtMonthlyInvestment.Text);
31             decimal yearlyInterestRate = Convert.ToDecimal(txtInterestRate.Text);
32             int years = Convert.ToInt32(txtYears.Text);
33
34             int months = years * 12;
35             decimal monthlyInterestRate = yearlyInterestRate / 12 / 100;
36
37             decimal futureValue = this.CalculateFutureValue(
38                 monthlyInvestment, monthlyInterestRate, months);
39             txtFutureValue.Text = futureValue.ToString("c");
40             txtMonthlyInvestment.Focus();
41         }
42     }
43     // added catch for format exception
44     catch (FormatException)
45     {
46         MessageBox.Show("A format exception has occurred, Please check all entries.",
```

Future Value

Monthly Investment: 100

Yearly Interest Rate:

Number of Years: 10

Future Value:

Calculate Exit

Entry Error

Investment Rate is a required field.

OK

Future Value

Monthly Investment: f

Yearly Interest Rate: 10

Number of Years: 10

Future Value:

Calculate Exit

Entry Error

Monthly Investment must be a decimal value.

OK

Future Value

Monthly Investment: 1500

Yearly Interest Rate: 10

Number of Years: 10

Future Value:

Calculate Exit

Entry Error

Monthly Investment must be between 1 and 1000.

OK

Future Value

Monthly Investment: 100

Yearly Interest Rate: 5

Number of Years: 10

Future Value: \$15,592.93

Calculate Exit

Visual Studio interface showing the code for `FutureValue` application. The code is in `Form1.cs` and implements the `btnCalculate_Click` event handler.

```
20 * Andrea Griffis
21 * 5/8/2020
22 * Exercise 7-2 Enhance the Future Value application
23 * *****/
24 1 reference
25 private void btnCalculate_Click(object sender, EventArgs e)
26 {
27     try
28     {
29         if (IsValidData()) // modified this event handler to validate data before it processes it
30         {
31             decimal monthlyInvestment = Convert.ToDecimal(txtMonthlyInvestment.Text);
32             decimal yearlyInterestRate = Convert.ToDecimal(txtInterestRate.Text);
33             int years = Convert.ToInt32(txtYears.Text);
34
35             int months = years * 12;
36             decimal monthlyInterestRate = yearlyInterestRate / 12 / 100;
37
38             decimal futureValue = this.CalculateFutureValue(
39                 monthlyInvestment, monthlyInterestRate, months);
40             txtFutureValue.Text = futureValue.ToString("c");
41             txtMonthlyInvestment.Focus();
42         }
43     }
44     /* added catch for format exception
45     catch (FormatException)
46     {
47         MessageBox.Show("A format exception has occurred, Please check all entries.",
48             "Format Entry Error");
49     }
50 }
```

The interface includes the following components:

- Menu Bar:** File, Edit, View, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q).
- Toolbox:** Contains various UI controls like Button, TextBox, etc.
- Solution Explorer:** Shows the project structure with `FutureValue` and `Form1`.
- Properties Window:** Shows the properties of the selected control.
- Output Window:** Displays the execution output.
- StatusBar:** Shows the current file (`FutureValue`), branch (`master`), and status (`Ready`).